

# Gedistribueerde systemen ontwerpen voor deployment

## DISTRIBUTED SYSTEM DESIGNERS

De Distributed System Designers zijn tools ter ondersteuning van het grafische design en de validatie van gedistribueerde systemen. Deze toolset bevat tools voor applicatiearchitecten, ontwerpers, ontwikkelaars en operations-architecten. De Distributed System Designers zijn een nieuw product van het Dynamic Systems Initiative (DSI), bedoeld om de ontwikkeling, het uitrollen en het beheer van gedistribueerde systemen bij bedrijven te verbeteren. Deze tools maken deel uit van Visual Studio 2005 Team System.

Servicegeoriënteerde architecturen vormen de basis van de volgende generatie gedistribueerde applicaties. Het Microsoft 'Indigo'-platform zal voorzien in een industry-leading implementatie van servicegeoriënteerde systemen. 'Indigo' bouwt voort op de ondersteuning van SOAP- en webservices die het Windows-platform tegenwoordig biedt, en voegt support toe aan een uitgebreide reeks transport- en systeemtopologieën waardoor veilige en betrouwbare op berichten gebaseerde communicatie tussen services mogelijk wordt. Het Indigo-platform biedt grote voordelen. Bedrijven kunnen nu servicegeoriënteerde systemen ontwikkelen met behulp van SOAP, XML-messaging en ASP.NET webservices. In dit artikel behandelen wij de tools voor het bouwen van gedistribueerde applicaties. Het artikel is wel gebaseerd op de bètaversie van Visual Studio Team System.

### Het ontwerp van gedistribueerde systemen

Het ontwerpen en uitrollen van gedistribueerde systemen is tamelijk gecompliceerd. Er is een aantal mogelijke problemen waar we tegen aan lopen. Visualisatie van de systeemstructuur als een integraal geheel wordt steeds moeilijker naarmate het systeem in een servicegeoriënteerde architectuur gefragmenteerd wordt. Bovendien hebben bedrijven in de loop der tijd vaak veel verschillende systemen verzameld, doordat de diverse afdelingen uiteenlopende applicaties hebben ingekocht of ontwikkeld. Aangezien de individuele programma's van verschillende programmeertechnologieën gebruikmaken, is het vaak moeilijk om functionaliteit en gegevens onderling te delen. Om samenwerking mogelijk te maken wordt van ontwikkelaars en architecten steeds meer verwacht dat zij berichtgebaseerde interfaces ontwerpen. Het ontwerpen van nieuwe berichten die overeenkomen met bestaande berichtenschema's is cruciaal. Samenwerking via berichten is het hart van een servicegeoriënteerde architectuur. Om de documentatie met betrekking tot het systeemontwerp actueel te houden dienen architecten en ontwikkelaars goed met elkaar te communiceren. Maar zelfs met de beste bedoelingen raakt de documentatie vaak verouderd en onnauwkeurig zodra ontwikkelaars beginnen met coderen. Het synchroniseren van ontwerpdocumenten met de snel wisselende programmatuur wordt een hels karwei dat al snel wordt opgegeven.

### Ontwerpen voor deployment

Leveranciers van software en hardware gaan er maar al te vaak van uit dat ontwikkelaars bekend zijn met de kleinste nuances van platformconfiguratie (SQL, IIS, BizTalk, enzovoort), en dat het operations management alles weet van frameworks en berichtpro-

tocollen die bij de ontwikkeling van applicaties worden gebruikt. Hoewel operations eigenlijk ook een stem zou moeten hebben in de ontwikkeling van software, wordt deze groep zowel organisatorisch als functioneel buiten de ontwikkeling gehouden. Operations en ontwikkelaars werken maar zelden proactief samen en vaak is het pas laat in de ontwikkelingscyclus dat ze bij elkaar komen om problemen vast te stellen die veel eerder hadden kunnen worden voorkomen. Een simpel voorbeeld: het ontwikkelen en het uitrollen van een webservice. De belangrijkste zorg van de ontwikkelaars is de implementatie van de service. Zij houden zich stuk voor stuk bezig met beveiliging en authenticatiemodellen, welke andere ondersteunende services in de doelomgeving nodig zijn en welke configuratie-instellingen het mogelijk maken dat de webservice werkt als gewenst. Operations wil weten welke protocollen en diensten voor de nieuwe service nodig zijn en of het IT-beleid van het bedrijf is nageleefd. Dit verschil tussen ontwikkelaars en operations leidt vaak tot problemen bij uitrollen. Dit heeft weer een verkeerde configuratie tot gevolg of erger nog, een design dat niet compatibel is met het computercentrum. Hierdoor gaat weer kostbaar IT-budget verloren om deze problemen op te lossen. Vele organisaties proberen dit communicatieprobleem te ondervangen met documentatie, design reviews en uitgebreide schema's, maar het ontbreekt hun aan tools en een gemeenschappelijke taal om hun beleid effectief op te leggen en na te leven. Bovendien staan deze processen zo ver af van de werkelijke tools die ontwikkelaars en operations dagelijks gebruiken dat het probleem alleen nog maar groter wordt.

### Applicaties configureren voor beveiliging

Het beveiligen van een gedistribueerde applicatie is een tijdrovend en ingewikkeld proces omdat het daarbij draait om talrijke technologieën en instellingen die van invloed kunnen zijn op het ontwerp van een applicatie. Er is momenteel geen geïntegreerde manier om de beveiligingsconfiguratie of de beveiligingseisen van het computercentrum al tijdens het ontwerpen van de applicatie weer te geven. Daardoor kan niet gegarandeerd worden dat de beveiliging juist wordt geïmplementeerd.

### Ontwerp en uitrollen verbeteren

De Distributed System Designers bieden een geïntegreerde ontwerp-omgeving met de doelstelling het visuele ontwerp en de validatie van gedistribueerde systemen mogelijk te maken. Ontwerpers gebruiken het System Definition Model (SDM) als het onderliggende metamodel dat niet alleen de connectivity, configuratie en relaties beschrijft voor de applicatieservices, maar ook voor de

runtime-omgevingen. SDM is gebaseerd op een model met meer lagen waarvan applicaties, omgevingen voor applicatiehosting, netwerktopologie, besturingssystemen en fysieke apparatuur deel uitmaken. Met dit model kunnen de Distributed System Designers niet alleen het ontwerp op iedere laag beschrijven, maar ook op iedere laag beperkingen en regels benoemen die over alle lagen van een gedistribueerd systeem doorgevoerd kunnen worden.

De Distributed System Designers ondersteunen een geïntegreerd model met twee domeinen: ontwikkeling en operations. Hierdoor kunnen de ontwerpers problemen van klanten op de volgende manieren behandelen:

- Een gemeenschappelijke taal (gebaseerd op SDM) om het ontwerp en de configuratie van een gedistribueerd systeem te beschrijven.
- Stelt ontwikkelaars in staat aan te geven welke eisen de applicatie aan de runtime-omgeving stelt.
- Stelt operations in staat aan te geven aan welke eisen de applicatie qua runtime, beveiliging en connectivity moet voldoen met het oog op de omgeving waar deze zal worden gebruikt.
- Maakt gebruik van abstracties waarmee ontwikkelaars en operations diensten in begrijpelijke taal kunnen communiceren.
- Integreert met het bestaande Visual Studio-projectstelsel en .NET-technologieën.
- Volledige synchronisatie tussen elementen van visueel design en code.
- Bevat een framework voor uitbreidingen waarmee nieuwe soorten applicatie en hostingsystemen kunnen worden vormgegeven.

## Overzicht van de Distributed System Designers

De Distributed System Designers omvatten de volgende designers:

- Application Connection Designer
- Logical Datacenter Designer
- System Designer
- Deployment Designer

## Application Connection Designer

De Application Connection Designer (ACD) helpt een ontwikkelaar of architect om applicaties te definiëren en configureren die worden samengesteld voor operationele systemen. De toolbox bevat een set vooraf gedefinieerde prototypen van applicaties, zoals webservices, webapplicaties, externe database, externe webservices en externe BizTalk-services. Ook is hierin een generiek prototype van een applicatie opgenomen. In het application connection diagram staan de applicaties die binnen een enkele Visual Studio-oplossing zijn gedefinieerd. Het diagram kan van het begin af aan worden gemaakt door er een applicatie vanuit de toolbox aan toe te voegen, of omgekeerd vanuit een bestaande oplossing of vanuit

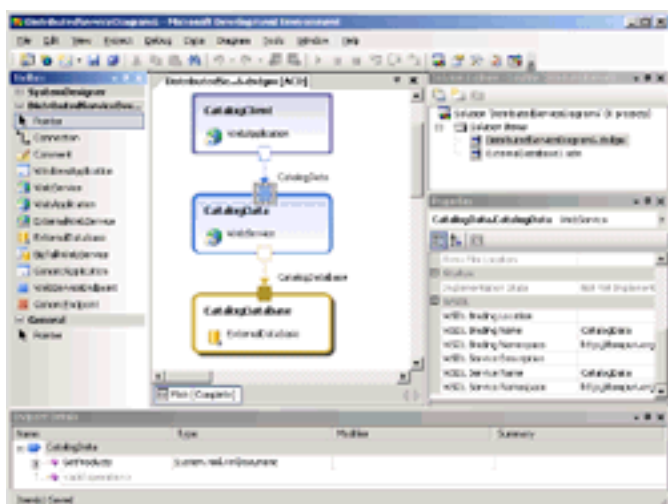
bestaande projecten. Een voorbeeld van een application connection diagram zie je in afbeelding 1.

De ACD geeft zowel 'interne' als 'externe' applicaties weer. Interne applicaties zijn individueel te gebruiken applicaties die worden gedefinieerd binnen de oplossing waaruit één of meer bruikbare systemen kunnen worden samengesteld. Iedere interne applicatie resulteert in een enkel project dat een uitvoerbare executable voor een bepaalde omgeving beschrijft (zoals een webapplicatie of Windows-applicatie). Resources kunnen in dit project worden opgenomen of via referenties verwezen worden naar andere projecten, waaronder class library-projecten. In externe applicaties kunnen ontwikkelaars referenties naar andere systemen visualiseren die tijdens het uitrollen bepaald moeten worden. Iedere applicatie wordt weergegeven als een blok met kleinere vormen die verbindingpunten voorstellen, hetzij van of naar de applicatie. Dit worden endpoints genoemd. Services die door de applicatie worden geboden, worden voorgesteld door provider-endpoints (vaste vormen); verbindingpunten naar services die worden aangeboden door andere applicaties, worden afgebeeld als consumer-endpoints (transparante vormen). Applicaties worden onderling verbonden via hun endpoints.

Het SDM-model is gericht op het type, de configuratie en verbinding van endpoints. Afzonderlijke applicaties kunnen dit model uitbreiden met de definitie van het gedrag van een service. Endpoints van een webservice kunnen bijvoorbeeld op twee manieren worden gedefinieerd. Als je op een provider-endpoint van een webservice klikt, verschijnt de definitie van activiteiten voor de desbetreffende webservice in het venster met Endpoint-details. Je kunt de definitie van het gedrag ook importeren uit een bestaand Web Service Description Language (.wsdl)-bestand. De verbindingen die op de ACD worden weergegeven, geven de huidige configuratie weer van de applicaties in de ontwikkelomgeving. Wanneer de applicatie op fouten wordt gecontroleerd, volgt deze de weergegeven verbindingsspaden.

De ACD ondersteunt een uitgestelde implementatie, zodat een gebruiker een ontwerp kan maken en valideren voordat hij het ontwerp in code vastlegt. Wanneer je vanuit de toolbox een applicatiedefinitie aan de ACD toevoegt, worden de overeenkomstige projecten, code en configuratie niet direct aangemaakt. De eerste generatie van codes wordt 'implementatie' genoemd. Applicatiedefinities kunnen stapsgewijs of allemaal tegelijk worden geïmplementeerd. Een geïmplementeerde applicatie is herkenbaar aan een slagschaduw in het diagram. Zodra een individuele applicatie is geïmplementeerd, worden de aangemaakte codebestanden en configuratiebestanden, en het in het diagram weergegeven applicatie-design continu gesynchroniseerd als het diagram en de codebestanden geopend zijn. Is het diagram om wat voor reden dan ook gesloten, dan wordt het weer geopend, met de gesynchroniseerde en bijgewerkte code inclusief de wijzigingen die in de code zijn aangebracht toen het diagram gesloten was. Op deze manier blijft het ontwerpmodel in de ACD altijd up-to-date als er wijzigingen in de code zijn aangebracht, en dat is een veel voorkomend probleem bij andere tools. Indien de applicatie is gemaakt door middel van reverse engineering van een bestaand project of een bestaande oplossing, wordt deze als reeds geïmplementeerd beschouwd en wordt deze vervolgens automatisch gesynchroniseerd.

Dankzij de mogelijkheid de implementatie uit te stellen kunnen architecten en applicatie-designers zich concentreren op het functionele ontwerp en de validatie van het systeem. Ze kunnen beslissingen uitstellen over implementatie die beter later door het ontwikkelteam kunnen worden genomen. Dit is bijvoorbeeld van toepassing op de keuze van de programmeertaal en de template of de keuze van de locatie van de server die voor webprojecten wordt gebruikt. Omdat het diagram als een bestand (.dsgm file)



Afbeelding 1. Application Connection Designer

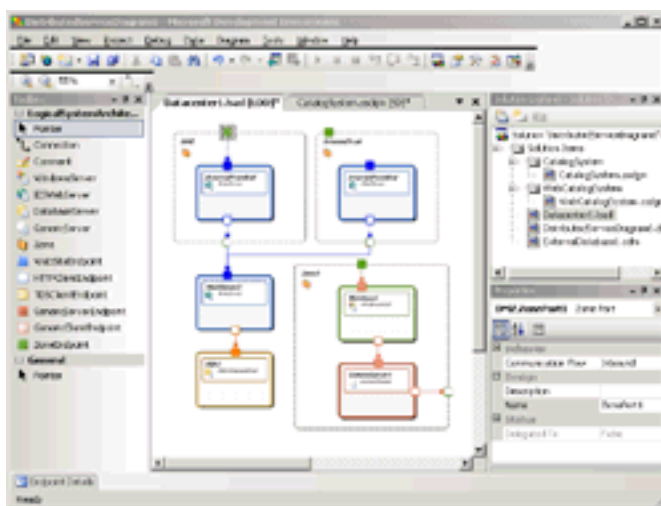
wordt opgeslagen, kan het worden opgenomen in de besturing van de broncode en in de normale werkstroom van een team. Informatie over geïmplementeerde applicaties wordt opgeslagen in .sdm-bestanden binnen het project. Projecten kunnen daardoor opnieuw worden gebruikt in meer oplossingen en het werk kan worden verdeeld onder de ontwikkelaars in een team. De ACD geeft toegang tot een volledig model met applicatie-instellingen en de mogelijkheid om beperkingen in een hosting-omgeving te definiëren. Deze hosting-beperkingen stellen de applicatie-designers in staat eisen te specificeren voor de hosting-omgeving. Door het definiëren van hosting-beperkingen voor applicatiedefinities in de Settings and Constraints Editor kunnen applicatie-designers aangeven dat er een verplicht aantal functies in de gebruiksomgeving aanwezig moet zijn. Zo stellen zij operations op de hoogte van belangrijke applicatie-eisen. Operations kan dan besluiten of zij wijzigingen willen aanbrengen in het logische computercentrum om aan deze eisen te voldoen. De instellingen van de applicatie en hosting-beperkingen worden gevalideerd tegen de instellingen en beperkingen van het computercentrum als onderdeel van de deployment-validatie (zie Deployment Designer).

### Logical Datacenter Designer

Met de Logical Datacenter Designer (LDD) worden diagrammen gemaakt van onderling verbonden logische servers die de logische structuur van een computercentrum vormen. Deze diagrammen van het logische computercentrum geven aan de ontwikkelaar belangrijke informatie door over de beoogde gebruiksomgeving. Met behulp van deze designer kunnen operations-architecten de types server in het computercentrum, de types toegestane communicatie, de specifieke communicatiepaden en types geactiveerde services specificeren en configureren.

De diagrammen van logische computercentra worden normaal gesproken gemaakt door en zijn eigendom van operations-analisten, maar worden gebruikt door ontwikkelaars. Als een operations-analist een diagram van een logisch computercentrum heeft gemaakt, kan hij dit diagram vergrendelen en de diverse versies van het diagram tijdens de ontwikkeling van de applicatie opslaan zodat wijzigingen in het ontwerp van het computercentrum worden doorgevoerd. Net als de andere Distributed System Designers is de Logical Datacenter Designer volledig geïntegreerd in Visual Studio. Maar diagrammen van logische computercentra worden los van de applicatieontwikkeling gemaakt. Deze diagrammen worden opgeslagen als .lsad-bestanden en kunnen opnieuw worden gebruikt door andere gedistribueerde systemen. In afbeelding 2 zie je een voorbeeld van een eenvoudig diagram van een logisch computercentrum.

De LDD bevat een toolbox met vooraf gedefinieerde prototypes van logische servers die aan het diagram kunnen worden toegevoegd.



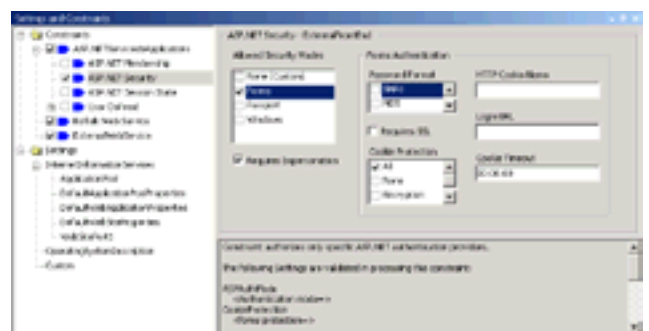
Afbeelding 2. Logical Datacenter Designer

Logische servers staan voor applicatie-hosts in het computercentrum. Native types zijn bijvoorbeeld een Windows Client-server, Webserver (IIS-server), SQL Server en een generieke server. Met uitzondering van de generieke server definieert ieder prototype een groep instellingen en beperkingen waarmee de server wordt geconfigureerd. De generieke server dient alleen voor documentatie en maakt het mogelijk dat er andere types server bestaan in het computercentrum. Iedere logische server heeft twee endpoints die een specifiek communicatieprotocol specificeren. Endpoints kunnen aan logische servers worden toegevoegd zodat deze via dat endpoint met andere logische servers kunnen communiceren. Door bewerking van de verstrekte instellingen kan de configuratie van logische servers in het computercentrum worden gemodelleerd. In plaats van het bewerken van de instellingen kunnen de instellingen ook vanaf bestaande servers worden geïmporteerd. Zodra het computercentrum is beschreven, kunnen beleidsmatige beperkingen ten aanzien van de configuratie en types applicatie die in het computercentrum gehost kunnen worden, met behulp van de Settings and Constraints Editor worden opgegeven. Deze beperkingen worden afgezet tegen de instellingen in de applicatielaag. ASP.NET-beveiligingsinstellingen kunnen bijvoorbeeld worden beperkt tot applicaties die worden gehost op een webserver, of de versie van het .NET Framework dat vereist is voor applicaties die worden gehost op een Windows Client Server. Het vastzetten van bepaalde instellingen houdt in dat deze instellingen niet kunnen worden aangepast of overschreven door ontwikkelaars. De Settings and Constraints Editor is afgebeeld in afbeelding 3.

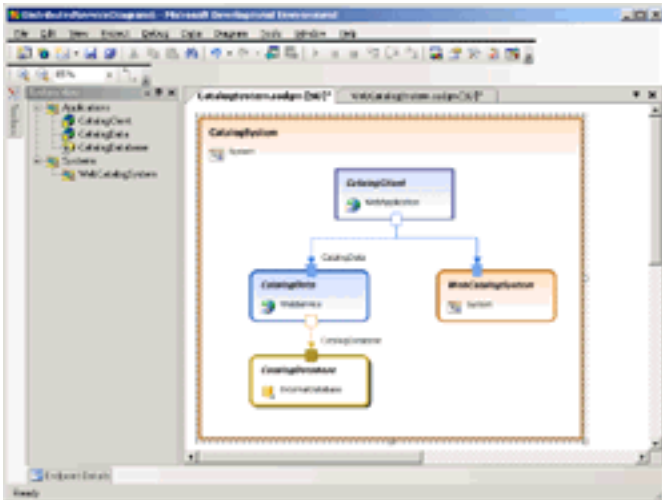
De toolbox bevat ook zones en endpoints. Zones zijn communicatiegrenzen in het computercentrum. De zone-endpoints zijn verbindingpunten tussen zones en servers. Communicatiepaden tussen zones en servers worden bestuurd door zone-endpoints. Deze kunnen worden geconfigureerd om te bepalen welk communicatieprotocol (ingaaend en uitgaand) worden toegestaan. Met server-endpoints worden communicatiepaden en -protocollen gespecificeerd die tussen servers zijn toegestaan en waarmee verkeer door de diverse zones wordt verzonden en ontvangen. De configuratie-instellingen en applicatiebeperkingen die in het diagram van het logische computercentrum zijn gedefinieerd, worden gevalideerd tegen de instellingen en beperkingen die zijn gedefinieerd voor applicaties als onderdeel van de deployment-validatie (zie Deployment Designer). Op die manier kunnen het beleid van het computercentrum en de eisen voor het uitrollen door de operationele teams naar de ontwikkelingsteams worden gecommuniceerd.

### System Designer

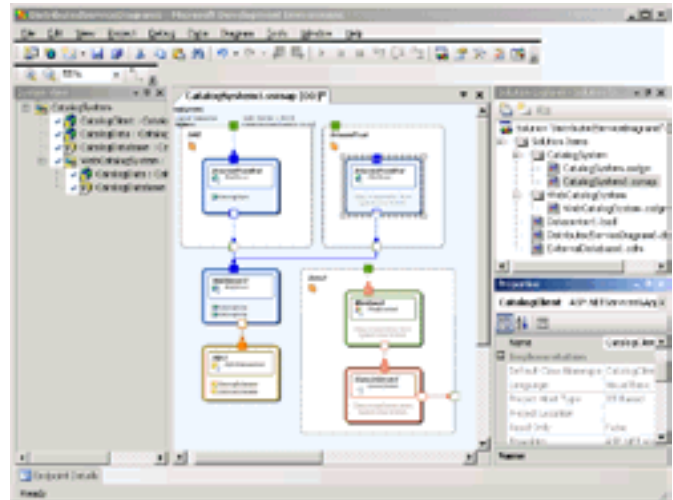
Met de System Designer worden systemen samengesteld en geconfigureerd op basis van applicaties die in de ACD zijn gedefinieerd. In de Distributed System Designers is een 'system' een eenheid van deployment en de configuratie van één of meer applicaties en andere subsystemen. Doordat systemen uit andere systemen kunnen worden samengesteld, kunnen scenario's van gedistribueerde systemen op grote schaal worden gedefinieerd. Een applicatiearchitect kan een complex, uit meer lagen bestaand systeem ontwerpen als een hiërarchie van 'geneste' systemen. Door een systeemdefini-



Afbeelding 3. Settings and Constraints Editor



Afbeelding 4. System Designer



Afbeelding 5. Deployment Designer

tie te maken is het mogelijk een uitrolconfiguratie te definiëren die onafhankelijk is van de ontwikkelingsconfiguratie weergegeven in de ACD. Indien toegestaan kunnen applicatie-instellingen worden overschreven bij de applicatie in een systeemdiagram. Er kunnen meer systeemdefinities worden aangemaakt, elk met verschillende configuraties van de applicaties die in de oplossing zijn gedefinieerd. Hierdoor kunnen verschillende configuraties worden gedefinieerd voor verschillende manieren van uitrollen; of voor verschillende computercentra, verschillende geografische gebieden of verschillende klanten. Een voorbeeld van een systeemdiagram met een genest systeem is afgebeeld in afbeelding 4.

## Deployment Designer

Met de Deployment Designer wordt bepaald hoe een specifiek systeem kan worden uitgerold in een logisch computercentrum. De Deployment Designer wordt met name gebruikt door ontwikkelaars en architecten. In de Deployment Designer worden applicaties in het systeem gekoppeld aan logische servers in het beoogde computercentrum. De functionaliteit van deze designer valt met name op door de communicatievoordelen die inherent zijn aan het SDM-model waarop de Distributed System Designers zijn gebaseerd. Zodra de deployment is gedefinieerd, kan dit op aanvraag worden gevalideerd. Validatie gaat na of alle applicaties in het systeem zijn gekoppeld aan logische servers, en controleert vervolgens of de applicaties voldoen aan de beperkingen die in het diagram van het logisch computercentrum zijn opgegeven. Validatie bevestigt ook dat de logische servers voldoen aan de hosting-beperkingen die zijn opgegeven in het diagram van de applicatieverbinding en in het systeemdiagram. Validatie garandeert ook dat de vereiste communicatiepaden aanwezig zijn, bepaalt of de juiste communicatieprotocollen gebruikt worden en dat ze compatibel zijn. Een voorbeeld van een gebruikdiagram is afgebeeld in afbeelding 5.

Alle validatiefouten worden weergegeven in de Visual Studio Error List. Wanneer je dubbelklikt op een fout in de Error List, opent de Deployment Designer het desbetreffende diagram, selecteert vervolgens de juiste applicatie of logische server en navigeert naar de desbetreffende instelling zodat deze gecorrigeerd kan worden. Op deze manier kun je configuratiefouten detecteren en corrigeren voor dat je gaat uitrollen. Vanuit de Deployment Designer kan een rapport met alle vereiste configuratie-instellingen van de applicatie en het computercentrum worden gemaakt.

## Uitbreidbaarheid

De Distributed System Designers zijn open en uitbreidbaar waardoor partners het platform kunnen uitbreiden en ondersteuning voor andere protocollen, extra soorten applicatie en logische servers kunnen toevoegen. De Distributed System Designers zijn een verzameling designers die het ontwerpen en valideren van gedistri-

bueerde systemen op basis van servicegeoriënteerde architecturen eenvoudiger maken.

- Servicegeoriënteerde architecturen: Van servicegeoriënteerde architecturen wordt verwacht dat zij fungeren als de basis van de volgende generatie gedistribueerde applicaties, maar het is moeilijk om hun noodzakelijkerwijs gefragmenteerde structuur te visualiseren en te ontwerpen. De Application Connection Designer en System Designer helpen architecten, ontwerpers en ontwikkelaars bij het visualiseren en ontwerpen van servicegeoriënteerde systemen als een geheel.
- Ontwerpen voor Operations: De Distributed System Designers tonen diagrammen van de applicatielaag en de hosting-laag van een systeem.. Deze diagrammen helpen de mensen die applicaties ontwerpen en moeten uitrollen bij het valideren van de configuratie tegen de operationele eisen wanneer de applicatie voor het eerst wordt ingezet, en tijdens de verdere levensduur als de desbetreffende diensten zich verder ontwikkelen.
- Uitvoerbaar ontwerp: Door het gecombineerde model in de Distributed System Designers kunnen applicatie- en infrastructuurarchitecten belangrijke metagegevens tijdens het ontwerpstadium vastleggen. Deze metagegevens kunnen gebruikt worden om code te genereren en het design van applicatie te valideren tegen de aanwezige services en beperkingen van het computercentrum. Met dit model zijn architecten in staat een formeler en rigouze ontwerp af te leveren. Het ontwerp loopt synchroon met de code en blijft relevant gedurende de hele levensduur van de applicatie, wat architecten helpt om de applicatie te onderhouden en uit te breiden waar nodig.

Ook helpen de Distributed System Designers ondernemingen om met succes servicegeoriënteerde applicaties op basis van webservices te bouwen en bereiden hen voor op de implementatie ervan in het computercentrum.

**Bill Gibson** is Program Manager in het Visual Studio Tools Team bij Microsoft Corporation. Hij is onder andere verantwoordelijk voor Logical Datacenter Designer en de Deployment Designer.

**Alex Torone** is Lead Program Manager in de Visual Studio Tools Team bij Microsoft Corporation. Hij heeft meer dan 20 jaar ervaring op het gebied van distributed computing, variërend van development van systemen voor vliegtuigen s tot commerciële softwareproducten voor Enterprise Applications.

### Nuttige internetadressen

Studio 2005 Team System:

<http://lab.msdn.microsoft.com/teamsystem/default.aspx>Meer informatie over DSI:

<http://www.microsoft.com/windowsserversystem/dsi>

Meer over Indigo: <http://msdn.microsoft.com/Longhorn/understanding/pillars/Indigo/>

Webcast: <http://msdn.microsoft.com/msdntv/episode.aspx?xml=episodes/en/20040129VSTUDIOAT/manifest.xml>